

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

AN IMPLEMENTATION METHODOLOGY AND SOFTWARE TOOL FOR AN ENTROPY BASED ENGINEERING MODEL FOR EVOLVING SYSTEMS

Matthew J. Behnke-DoD Civilian

B.S., Kettering University, 2001

Master of Science in Software Engineering-June 2003

Advisor: Man-Tak Shing, Department of Computer Science

Second Reader: Chris Miles, U.S. Army Tank Automotive Command

This thesis presents a practical method for calculating and representing entropy-based metrics for a set of bibliographic records evolving over time, in support of Dr. Michael Saboe's dissertation research which addressed the ability to measure software technology transfer. The implementation of the analysis methodology for determining the information-temperature of evolving datasets containing bibliographic records is described. The information-temperature metric is based on information entropy and is used to relate the maximum complexity of a system to the current complexity.

The implementation of the analysis methodology required using data mining techniques to prepare the datasets. Additionally, since the information-temperature metric derived from Saboe's work was a new emerging concept, the data analysis methodology had to be refined several times in order to obtain the desired results. An iterative software development paradigm was used to write the application in three iterations using Visual Basic.

At the end of the implementation the data analysis process became systemized, allowing the outlining of the steps to compute the temperature of datasets, and it is estimated that the learning curve of the analysis can be reduced by 50 percent through integration and packing of the analysis functions into a stand-alone application with an intuitive user interface.

KEYWORDS: Software Engineering, Entropy, Information Theory, Software Methodology, Data Analysis, Data Mining, Bibliographic, Tech-OASIS, Vantagepoint, Technology Transfer, Iterative Development, Information Temperature, DataThermometer, Saboe Degrees

EXTENDING THE COMPUTER-AIDED SOFTWARE EVOLUTION SYSTEM (CASES) WITH QUALITY FUNCTION DEPLOYMENT (QFD)

Arthur B. Clomera-Major, United States Army

B.S., University of California-Davis, 1991

Master of Science in Software Engineering-June 2003

Advisor: Man-Tak Shing, Department of Computer Science

Second Reader: LTC Joseph F. Puett, III, USA

This thesis extends the Computer Aided Software Evolution System (CASES) with Quality Function Deployment (QFD) to enhance dependency traceability (type and degree) between software development artifacts. Embedding Quality Function Deployment (QFD) in the Relational Hypergraph Software Evolution Model to prototype a Holistic Framework for Software Engineering (HFSE) is the major task achieved by this thesis. CASES is implemented by using Java Development Kit (JDK) 1.3.1 and an open software architecture. The primary contributions of this research include: 1) embedding QFD into CASES to record and track artifact dependencies, 2) providing engineering views of QFD dependencies, and 3) providing a stakeholder Graphical User Interface (GUI) to define and manage any software evolution process.

These major contributions allow a software engineer to: 1) input, modify, and analyze dependency characteristics between software artifacts within a QFD framework, 2) make decisions based upon views of dependency information, and 3) design a custom software evolution model through the use of a GUI.

SOFTWARE ENGINEERING

KEYWORDS: Software Engineering, Software Evolution, Integrated Software Development Environments, Software Quality Function Deployment

NEXT GENERATION SOFTWARE PROCESS IMPROVEMENT

Daniel Turnas-DoD Civilian, United States Army

B.S., University of Michigan, 1999

Master of Science in Software Engineering-June 2003

Advisor: Mikhail Auguston, Department of Computer Science

Second Reader: Chris Miles, U.S. Army Tank Automotive Command

Software is often developed under a process that can at best be described as ad hoc. While it is possible to develop quality software under an ad hoc process, formal processes can be developed to help increase the overall quality of the software under development. The application of these processes allows for an organization to mature. The software maturity level, and process improvement, of an organization can be measured with the Capability Maturity Model. The scope of this work is to use organizationally improved software processes on a small scale software product developed by the U.S. Army. The goal is to establish process improvement based on the Capability Maturity Model.

KEYWORDS: CMM, Process Improvement, Software Engineering, Requirements Management, Risk Management, Software Design